

REDACTED AND PUBLICLY FILED

**UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

SAS INSTITUTE INC.,

Plaintiff,

vs.

WORLD PROGRAMMING LIMITED *et al.*,

Defendants.

Case 2:18-cv-00295-JRG

**OPENING BRIEF IN SUPPORT OF PLAINTIFF'S POSITION FOR
COPYRIGHTABILITY HEARING**

REDACTED AND PUBLICLY FILED

TABLE OF CONTENTS

	<u>Page</u>
TABLE OF AUTHORITIES	II
I. INTRODUCTION	1
II. THE SAS SYSTEM IS PROTECTED BY COPYRIGHT.....	1
A. The Asserted Work is the SAS System.....	1
B. Copyrights for the SAS Manuals Provide Additional Protection for the SAS System.....	3
III. COPYRIGHT LAW GENERALLY.....	5
IV. THE ABSTRACTION-FILTRATION-COMPARISON PROCESS	6
A. The Abstraction-Filtration-Comparison Analysis.....	6
B. Professor Storer's Expert Report	11
C. Creativity and Expressiveness in the SAS System.	13
D. WPL's Open and Notorious Copying of the SAS System.....	17
V. WPL'S ABSTRACTION-FILTRATION PROCESS IS UNAVAILING.	21
A. The Burden of Further Filtration is on WPL.	21
B. WPL's Filtration Analysis Is Wrong.	23
1. Public Domain	23
2. Merger.....	24
3. <i>Scènes à faire</i>	25
4. Short Phrases.....	25
VI. JUDICIAL RESOURCES MAY BE WASTED IF THE COURT OR JURY RESOLVES ISSUES CONCERNING SAS'S INPUT FORMATS BEFORE THE SUPREME COURT ANNOUNCES ITS OPINION IN <i>GOOGLE LLC V. ORACLE AMERICA INC.</i>	27

REDACTED AND PUBLICLY FILED

TABLE OF AUTHORITIES

Cases

<i>Aspen Tech., Inc. v. M3 Tech., Inc.</i> , 569 F. App'x 259 (5th Cir. 2014).....	30
<i>Atari Games Corp. v. Nintendo of Am., Inc.</i> , 975 F.2d 832 (Fed.Cir.1992).....	5, 24, 26, 27
<i>Castle Rock Ent't, Inc. v. Carol Pub. Grp., Inc.</i> , 150 F.3d 132 (2d Cir. 1998)	2
<i>Cisco Sys., Inc. v. Arista Networks, Inc.</i> , 5:14-cv-05344-BLF, Doc. 633, 635 (N.D. Cal. Nov. 10, 2016).....	2
<i>Compaq Computer Corp. v. Ergonome, Inc.</i> , 137 F.Supp.2d 768 (S.D. Tex. 1981)	27
<i>Compulife Software Inc. v. Newman</i> , 959 F.3d 1288 (11th Cir. 2020).....	21-22
<i>Computer Assocs. Int'l v. Altai, Inc.</i> , 982 F.2d 693 (2d Cir. 1992).....	6
<i>Computer Management Assistance Co. v. Robert F. DeCastro, Inc.</i> , 220 F.3d 396 (5th Cir. 2000).....	5
<i>Eldred v. Ashcroft</i> , 537 U.S. 186 (2003).....	1-2
<i>Eng'g Dynamics, Inc. v. Structural Software, Inc.</i> , 26 F.3d 1335 (5th Cir. 1994).....	2, 3-4, 6, 9, 29-30
<i>Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.</i> , 499 U.S. 340 (1991)	5-6
<i>Fourth Est. Pub. Benefit Corp. v. Wal-Street.com, LLC</i> , 139 S. Ct. 881 (2019).....	1, 2
<i>Gates Rubber Co. v. Bando Chem. Indus., Ltd.</i> , 9 F.3d 823 (10th Cir. 1993).....	30
<i>Google LLC v. Oracle America Inc.</i> , No. 18-956.....	27-28
<i>J. Racenstein & Co. v. Wallace</i> , 1999 WL 632853 (S.D.N.Y. Aug. 19, 1999)	26
<i>Markman v. Westview Instruments, Inc.</i> , 517 U.S. 370 (1996)	30
<i>Mitel, Inc. v. Iqtel, Inc.</i> , 124 F.3d 1366 (10th Cir. 1997).....	9
<i>N. Coast Indus. v. Jason Maxwell, Inc.</i> , 972 F.2d 1031 (9th Cir. 1992).....	30
<i>Oracle Am., Inc. v. Google Inc.</i> , No. 3:10-cv-03561-WHA, Doc. 1089 (N.D. Cal. May 7, 2012)	2

REDACTED AND PUBLICLY FILED

<i>Oracle Am., Inc. v. Google v. Google, Inc.</i> , 750 F.3d 1339 (Fed. Cir. 2014) <i>passim</i>	
<i>SAS Inst. Inc. v. S & H Computer Systems, Inc.</i> , 605 F.Supp. 816 (M.D. Tenn. 1985).....	3, 23-24
<i>SAS Inst. Inc. v. World Programming Ltd.</i> , 952 F.3d 513 (4th Cir. 2020)	3
<i>Sid & Marty Krofft Tele. Prods. v. McDonald's Corp.</i> , 1983 WL 1142 (C.D. Cal. Jan. 12, 1983) 2	
<i>United States v. Gaudin</i> , 515 U.S. 506 (1995)	30
<i>Ventures Educ. Sys. Corp. v. Prof'l Dev. Assocs., Inc.</i> , 2008 WL 316667 (S.D.N.Y. Jul. 31, 2008)	27
<i>Warner Bros. Ent't, Inc. v. RDR Books</i> , 575 F. Supp. 2d 513 (S.D.N.Y. 2008)	2
Statutes	
17 U.S.C. § 101.....	5
17 U.S.C. § 102.....	22
17 U.S.C. § 102(a)	5
17 U.S.C. § 102(b).....	9
17 U.S.C. § 106.....	2
17 U.S.C. § 408(a)	1
Other Authorities	
4 Nimmer on Copyright § 13.03[F][3]	22
H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54, <i>reprinted in</i> 1976 U.S.C.C.A.N. 5659.....	5
<i>The Best for Last: The Timing of U.S. Supreme Court Decisions</i> , 64 Duke L.J. 991, 993 (2015) 28	

REDACTED AND PUBLICLY FILED

I. INTRODUCTION

In accordance with the Court’s Order As To Copyrightability issued on August 24, 2020 (ECF No. 436), Plaintiff SAS Institute Inc. (“SAS”) respectfully submits its opening brief which sets forth SAS’s position on the subject.

In an effort to streamline the case in a manner consistent with the Court’s comments at the August 24, 2020 hearing, SAS, as described below, has further limited its copyright claim in an effort to facilitate the Court’s determination of the “core protectable expression” at issue in this case. (*See* ECF No. 436, at 2.)

II. THE SAS SYSTEM IS PROTECTED BY COPYRIGHT

A. The Asserted Work is the SAS System.

As SAS explained earlier (ECF No. 264), WPL itself in internal documents describes how its commercial proposition was to create a “clone” of the SAS System software in order to offer its copy – WPS – to SAS users at a lower price. (*Id.* at 14-20.) The SAS System is a constantly evolving software program that SAS has been continually updating and improving for 44 years. (ECF No. 264-1 (“Collins Decl.”), at ¶ 17.) As explained by Keith Collins, SAS’s former Chief Technical Officer, as SAS has released new versions of its software over the course of four decades, SAS has continuously added new elements to the SAS System and expanded elements present in earlier versions. (*Id.* at ¶ 16.)

Copyright protection for the SAS System (including all of its creative input formats and output designs) is automatic under the law upon creation, and is not tied to copyright registrations for the SAS System. *See Fourth Est. Pub. Benefit Corp. v. Wal-Street.com, LLC*, 139 S. Ct. 881, 887 (2019) (An “author gains ‘exclusive rights’ in her work immediately upon the work’s creation.”); *Eldred v. Ashcroft*, 537 U.S. 186, 195 (2003) (“[F]ederal copyright … protection run[s] from the work’s creation.”). Copyright “registration is not a condition of copyright protection.”

REDACTED AND PUBLICLY FILED

See 17 U.S.C. § 408(a). Instead, registration is “akin to an administrative exhaustion requirement that the owner must satisfy before suing to enforce ownership rights.” *Fourth Estate*, 139 S. Ct. at 887.

For purposes of conducting an Abstraction-Filtration-Comparison analysis, a software program like the SAS System, with multiple updated versions, should be treated as a single unified work. *See Castle Rock Ent’t, Inc. v. Carol Pub. Grp., Inc.*, 150 F.3d 132, 138 (2d Cir. 1998) (84 episodes of *Seinfeld* treated as a single work where accused work focused on the entire continuous television series)¹; *Warner Bros. Ent’t, Inc. v. RDR Books*, 575 F. Supp. 2d 513, 535 n.14 (S.D.N.Y. 2008) (treating *Harry Potter* series as a single work that tells “one coherent narrative”); *Sid & Marty Krofft Tele. Prods. v. McDonald’s Corp.*, 1983 WL 1142, at *5 (C.D. Cal. Jan. 12, 1983). The Fifth Circuit has taken a similar approach to software, focusing the analysis on copying of input and output formats from a suite of programs, not on a version-by-version analysis. *See Eng’g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1339-42 (5th Cir. 1994) (analyzing copyrightability of input/output formats and user interface based upon copyright registrations for four user manuals related to a suite of computer programs). Courts in other recent software cases, including *Cisco v. Arista* and *Oracle v. Google*, have followed suit.²

¹ The Second Circuit recognized that while 17 U.S.C. § 106 speaks in terms of a single copyrighted “work,” it would “elevate form over substance” to treat copying from a series of related programs as distinguishable from copying from a “combination of creative works that do constitute a discrete series of works.” *Castle Rock*, 150 F.3d at 138.

² See *Cisco Sys., Inc. v. Arista Networks, Inc.*, 5:14-cv-05344-BLF, Doc. 633, 635 (N.D. Cal. Nov. 10, 2016) (parties’ trial briefs regarding “copyrighted work”); *id.*, Doc. No. 737 (N.D. Cal. Dec. 12, 2016) (Jury Instruction Nos. 25 & 31 defining the “copyrighted works involved in this trial” as “Cisco’s four user interfaces” for each of Cisco’s operating systems and “Cisco’s technical manuals,” covered by “26 certificates of copyright registration”). *Oracle Am., Inc. v. Google Inc.*, No. 3:10-cv-03561-WHA, Doc. 1089 (N.D. Cal. May 7, 2012) (Special Verdict Form treating copyrighted work as the “compilable code for the 37 Java API packages in question taken as a group”); *Oracle Am., Inc. v. Google, Inc.*, No. 3:10-cv-03561-WHA, 2012 WL 12924987, at *1-2 (N.D. Cal. May 16, 2012) (holding Oracle was the owner and copyright holder of the “asserted 37

REDACTED AND PUBLICLY FILED

SAS is the author of and owns all versions of the SAS System, at least from 1976 to the present. SAS's ownership of the SAS System is supported by a history of periodic registrations on the SAS System, which SAS obtained for numerous major releases, beginning with version 79.5 of the SAS System. *SAS Inst., Inc. v. S & H Computer Systems, Inc. v. SAS Inst., Inc.*, 605 F. Supp. 816, 818-19 (M.D. Tenn. 1985); Collins Decl. ¶ 19 & Ex. C (attaching copyright registrations.) SAS also obtained a copyright registration for the SAS Learning Edition, which WPL obtained a dozen copies of by defrauding SAS. *See SAS Inst. Inc. v. World Programming Ltd.*, 952 F.3d 513, 518-19 (4th Cir. 2020). The SAS Learning Edition gave WPL full access to the core elements of the SAS System software (limited to 1500 observations). It is a related, derivative work, of the SAS System and is not a separate asserted work from the SAS System for purposes of conducting the Abstraction-Filtration-Comparison analysis.

B. Copyrights for the SAS Manuals Provide Additional Protection for the SAS System.

In response to the Court's request that the parties further narrow the scope of this case, SAS has determined that it will limit its asserted work to the SAS System software. WPL, however, also gained access to the SAS System (and its input formats, output designs, and keywords) through numerous copyrighted SAS Manuals that provide thousands of pages of extensive, detailed examples of SAS output designs, and describe how SAS outputs may be achieved using the creative expression reflected in the input formats (*e.g.*, statements, options, and syntax) that SAS defined. As the Fifth Circuit recognized in *Engineering Dynamics*, creative inputs and outputs in a suite of software products may be further protected by copyrights on manuals that describe the asserted work. In *Engineering Dynamics*, the plaintiff did not copyright

API packages and eleven code files" by reference to two copyright registrations for Versions 1.4 and 5.0 of the Java 2 Standard Edition, which were registered as derivative works (or compilations) of prior works with the addition of new and revised computer code)).

REDACTED AND PUBLICLY FILED

any of the actual computer programs comprising the suites of software products at issue, but had “registered copyrights in four user manuals containing detailed verbal descriptions and pictorial representations of input and output formats.” *Id.* at 1340. Those user manual copyrights were sufficient to protect the input and output formats in Engineering Dynamics’ software. 26 F.3d at 1340. Accordingly, copyrights for the SAS Manuals provide additional copyright protection for the SAS System’s input formats and output designs recorded in the Manuals, though SAS does not rely on them as separate asserted works.



REDACTED AND PUBLICLY FILED



For purposes of narrowing and simplifying this case, SAS will not treat these publications as separate asserted works, but as copyrighted works that document (and provide further protection for) the creative expression in the SAS System software, and which provided WPL with further access for the purposes of copying the SAS System.

III. COPYRIGHT LAW GENERALLY

The Copyright Act provides protection to “original works of authorship fixed in any tangible medium of expression,” including “literary works.” 17 U.S.C. § 102(a). “Computer programs” – defined in the Copyright Act as “a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result,” 17 U.S.C. § 101 – can be subject to copyright protection as “literary works.” *See Computer Management Assistance Co. v. Robert F. DeCastro, Inc.*, 220 F.3d 396, 400 (5th Cir. 2000); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 838 (Fed.Cir.1992) (“As literary works, copyright protection extends to computer programs.”). The legislative history of the Copyright Act explains that “literary works” includes “computer programs to the extent that they incorporate authorship in the programmer’s expression of original ideas, as distinguished from the ideas themselves.” H.R.Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.

Under the Copyright Act, a work must be “original” to qualify for copyright protection. 17 U.S.C. § 102(a). The “originality requirement is not particularly stringent,” however. *Feist Publ’ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 358 (1991). “Original, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied

REDACTED AND PUBLICLY FILED

from other works), and that it possesses at least some minimal degree of creativity.” *Id.* at 345.

The law with respect to software in the Fifth Circuit is that “copyright protection extends not only to the literal elements of a program, *i.e.*, its source code and object code, but also to its ‘nonliteral’ elements, such as the program architecture, ‘structure, sequence and organization,’ operational modules, and computer-user interface.” *Eng’g Dynamics*, 26 F.3d at 1341; *see also Oracle America, Inc. v. Google, Inc.*, 750 F.3d 1339, 1355-56 (Fed. Cir. 2014) (“[W]hether the non-literal elements of a program are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself.”) (internal citation omitted). In addition to protection of a “computer-user interface,” *i.e.*, input formats, the same analysis applies to output designs, in other words, “some output formats will contain sufficient original expression to merit protection.” *Eng’g Dynamics*, 26 F.3d at 1342.

IV. THE ABSTRACTION-FILTRATION-COMPARISON PROCESS

A. The Abstraction-Filtration-Comparison Analysis.

In *Oracle America, Inc. v. Google Inc.*, 750 F.3d 1339 (Fed. Cir. 2014), the case most similar to this one at least as far as input formats are concerned, the Federal Circuit addressed the issue of the abstraction-filtration-comparison analysis with regard to the copying of non-literal elements of software. *Id.* at 1357. The court noted that the analysis had first been formulated in the Second Circuit and ultimately endorsed in the Ninth Circuit (whose law applied in *Oracle*.) *Id.* The analysis, according to the *Oracle* Court, “eschews bright line approaches and requires a more nuanced assessment of the particular program at issue in order to determine what expression is protectable and infringed.” *Id.* at 1357.

The court then outlined the “three steps” of the test as described by the Second Circuit in *Computer Assocs. Int’l v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), beginning with the first two:

In the abstraction step, the court “first break[s] down the allegedly infringed

REDACTED AND PUBLICLY FILED

program into its constituent structural parts.” *Id.* at 706. In the filtration step, the court “sift[s] out all non-protectable material,” including ideas and “expression that is necessarily incidental to those ideas.” *Id.*

750 F.3d at 1357.³

With regard to the filtration step, “the court is first to assess whether the expression is original to the programmer or author.” 750 F.3d at 1357. After that, filtration “then determine[s] whether the particular inclusion of any level of abstraction is dictated by considerations of efficiency, required by factors already external to the program itself, or taken from the public domain – all of which would render the expression unprotectable.” *Id.* at 1357-58. In undertaking that process, “conclusions are to be informed by traditional copyright principles of originality, merger, and scenes a faire.” *Id.* at 1358.

In *Oracle*, Oracle’s predecessor “wrote a number of ready-to-use Java programs to perform common computer functions and organized those programs into groups it called ‘packages.’” *Id.* at 1348-49. Those packages, 37 of which were at issue, “allow[ed] programmers to use the pre-written code to build certain functions into their own programs, rather than write their own code to perform those functions from scratch.” *Id.* at 1349. Oracle’s “packages” (also called APIs (application programming interfaces) by the court) “are shortcuts.” *Id.* at 1349.

The Oracle APIs comprised an “elaborately organized” set of “all the names of methods, classes, interfaces, and packages,” 750 F.3d at 1351, that represented the “structure, sequence, and organization” (SSO) of “the 37 packages.” *Id.* The court explained:

[Oracle] called the code for a specific operation (function) a “method.” It defined “classes” so that each class consists of specified methods plus variables and other elements on which the methods operate. To organize the classes for users, then, it grouped classes (along with certain related “interfaces”) into “packages.”

³ The final step, not yet at issue here, “compares the remaining creative expression with the allegedly infringing program.” 750 F.3d at 1357.

REDACTED AND PUBLICLY FILED

Id. at 1349. As the court put it, “Oracle’s collection of API packages is like a library, each package is like a bookshelf in the library, each class is like a book on the shelf, and each method is like a how-to chapter in a book.” *Id.* Each package, in turn, “contain[s] thousands of individual elements, including classes, subclasses, methods, and interfaces.” *Id.*⁴

The court defined “declaring code” as “the expression that identifies the prewritten function and is sometimes referred to as the ‘declaration’ or ‘header.’” 750 F.3d at 1349. The court explained: the “main point is that this header line of code introduces the method body and specifies very precisely the inputs, name and other functionality.” *Id.* (internal citation omitted). The “expressions used by the programmer from the declaring code command the computer to execute the associated implementing code, which gives the computer the step-by-step instructions for carrying out the declared function.” *Id.* In the court’s words, Google copied “all the names of methods, classes, interfaces, and packages—the overall system of organized names—covering 37 packages, with over six hundred classes, with over six thousand methods.” *Id.* at 1351 (internal quotation omitted).

⁴ The *Oracle* Court provided an example, albeit one much simpler than the SAS PROCs at issue here, from a package called “java.lang”:

Within that package is a class called “math,” and within “math” there are several methods, including one that is designed to find the larger of two numbers: “max.” The declaration for the “max” method, as defined for integers, is: “public static int max(int x, int y),” where the word “public” means that the method is generally accessible, “static” means that no specific instance of the class is needed to call the method, the first “int” indicates that the method returns an integer, and “int x” and “int y” are the two numbers (inputs) being compared. A programmer calls the “max” method by typing the name of the method stated in the declaring code and providing unique inputs for the variables “x” and “y.” The expressions used command the computer to execute the implementing code that carries out the operation of returning the larger number.

750 F.3d at 1349 (internal citation omitted).

REDACTED AND PUBLICLY FILED

The district court in *Oracle* first “acknowledged that the overall structure of Oracle’s API packages is creative [and] original.” 750 F.3d at 1352. After an infringement finding by the jury, the court found that the structure was “nevertheless a command structure, a system or method of operation – a long hierarchy of over six thousand commands to carry out pre-assigned functions – that [was] not entitled to copyright protection under Section 102(b) of the Copyright Act.” *Id.* (internal quotation omitted). As a result, the district court “found all aspects of the SSO devoid of protection as a ‘method of operation’ under 17 U.S.C. § 102(b).” *Id.* at 1359.

The Federal Circuit reversed. 750 F.3d at 1349 (“Because we conclude that the declaring code and the structure, sequence, and organization of the API packages are entitled to copyright protection, we reverse the district court’s copyrightability determination.). As an initial matter, the court found that “designing Java API packages was a creative process” because “the Sun/Oracle developers had a vast range of options for the structure and organization.” *Id.* at 1356. The Federal Circuit also rejected the district court’s overarching conclusion “that anything that performs a function is necessarily uncopyrightable.” *Id.* at 1357 (citing *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1372 (10th Cir. 1997)).

As noted, the Federal Circuit had already initially determined that the design of the packages was original based on the “vast range of options for the structure and organization.” 750 F.3d 1356; *see also Eng’g Dynamics*, 26 F.3d at 1346 (the “creativity inherent in [plaintiff’s] program is proved by the existence by other, dissimilar structural engineering programs available in the market.”). Indeed, *Eng’g Dynamics* also held that the “utilitarian function of the input formats, which ultimately act like switches in the electrical circuits of the program,” do not “outweigh their expressive purpose,” *id.* at 1346, and that “the formats, taken as a whole, readily qualify as ‘expression’ measured against the ideas versus expression dichotomy. *Id.* at 1344.

REDACTED AND PUBLICLY FILED

In the end, the Federal Circuit’s ruling in large part turned on the fact that Oracle claimed “copyright protection only in its *particular* way of naming and organizing each of the 37 Java API packages. 750 F.3d at 1367 (emphasis in original). The defendant, for its part, “could have structured [its product] differently and could have chosen different ways to express and implement the functionality that it copied. *Id.* at 1368.⁵ Google, for that reason, “like any author, is not permitted to employ the precise phrasing or precise structure chosen by Oracle to flesh out the substance of its packages – the details and arrangement of the prose.” *Id.* at 1368. Therefore, the district court’s finding of no copyrightability was reversed. *Id.* at 1381. Oracle’s parallels to this case are notable (although this case is ultimately much stronger because it has a further set of copied elements – in the form of output designs – not present in the Oracle case).

In a later case bearing similarities to this one (and following guidance from Oracle), the court in *Cisco Sys., Inc. v. Arista Networks, Inc.*, No. 14-cv-05344-BLF, slip op. (attached) (N.D. Cal. Dec. 9, 2016), considered the issue of an abstraction-filtration-comparison test in a case concerning copying of “text-based user interfaces” *id.* at 2, again similar to the input formats at issue here. As part of the abstraction step, the plaintiff identified “protectable elements in six categories or ‘building blocks’ as the sources of copyright infringement,” including input formats in the nature of “command expressions” and “modes and prompts.” *Id.* at 6. The court held that even though “virtually all of the individual elements of the asserted modes and prompts existed prior to [plaintiff’s] creation, the selection and arrangement of the identified modes and prompts can be claimed as original.” *Id.* at 14. The court also found the “building blocks” “protectable as

⁵ Indeed, other competitors did so. 750 F.3d at 1360 n. 5 (“It is undisputed that Microsoft and Apple developed mobile operating systems from scratch, using their own array of software packages.”). The competing statistical software here, such as SPSS and R, have markedly different input formats and output designs. (See Declaration of James A. Storer, filed herewith, at ¶¶ 21-23 & Ex. F.)

REDACTED AND PUBLICLY FILED

a compilation” and found that each of plaintiff’s “user interfaces as a whole is subject to protection as a compilation of those building blocks.” *Id.* at 18.

B. Professor Storer’s Expert Report

SAS’s expert, Professor Storer, provided an Expert Report in which he conducted an abstraction-filtration-comparison analysis. (Storer Decl., Ex. A at pp. 26-57.) In that Report, as summarized in his Declaration (at ¶¶ 6-31), Professor Storer first describes the manner in which he conducted the Abstraction step by breaking the SAS System in five levels of abstraction:

1. Main purpose of the program.
2. Interface mechanism.
3. Input formats (including the collection of PROCs, statements, options, formats, informats, global statements, access engines and other elements available to the user and the syntax, all of which dictate what the user’s input must look like).
4. Output designs (the collection of content and formatting, including default parameters, used to display information in response to the user’s input).
5. Naming and syntax of individual PROCs, statements, options, default parameters, and other elements.

(Storer Decl. ¶ 11.) As an initial matter, Professor Storer filtered out entirely the first two levels of abstraction. (*Id.* at ¶ 12.)

With regard to the third abstracted element – the collection of input formats in the SAS System – Professor Storer defined those as including “the collection of PROCs, statements, options, formats, informats, global statements, access engines and other elements, and the syntax governing how those PROCs and elements may be combined by the user and how they relate to each other.” (Storer Report at 27.) The fourth element – the collection of output designs in the SAS System – were defined as including “the order, types of statistical and graphical output, and other identification of output, including output designs resulting from default parameters.” (*Id.* at 28.) For the fifth element – the naming and syntax of individual elements within the SAS System

REDACTED AND PUBLICLY FILED

– Professor Storer described that it included the names of individual elements and the syntax used with those. (*Id.* at 28-30.)

Professor Storer’s Report, as a part of its filtration analysis, describes in detail the creative and expressive aspects of both the input formats and output designs in the SAS System. (See Storer Decl. ¶¶ 18-30.)

For the fifth element, Professor Storer used an example from SAS’s PROC MIXED, a procedure that fits different mixed linear models to data and enables users to make statistical inferences about the data. (Storer Report at 28.) PROC MIXED, which has been copied into WPS has the following structure as input format options as shown in the SAS manual:

SAS/STAT 14.1 User’s Guide: The MIXED Procedure

Product: SAS/STAT

Procedure: Proc Mixed

```
PROC MIXED <options>;
  BY variables ;
  CLASS variable <(REF= option)> ... <variable <(REF= option)>> </ global-option>;
  CODE <options>;
  ID variables ;
  MODEL dependent = <fixed-effects> </ options>;
  RANDOM random-effects </ options>;
  REPEATED <repeated-effect> </ options>;
  PARMS (value-list)... </ options>;
  PRIOR <distribution> </ options>;
  CONTRAST 'label' <fixed-effect values ... >
    <| random-effect values ... >, ... </ options>;
  ESTIMATE 'label' <fixed-effect values ... >
    <| random-effect values ... > </ options>;
  LSMEANS fixed-effects </ options>;
  LSMESTIMATE model-effect lsmeestimate-specification </ options>;
  SLICE model-effect </ options>;
  STORE <OUT=>item-store-name </ LABEL='label'>;
  WEIGHT variable ;
```

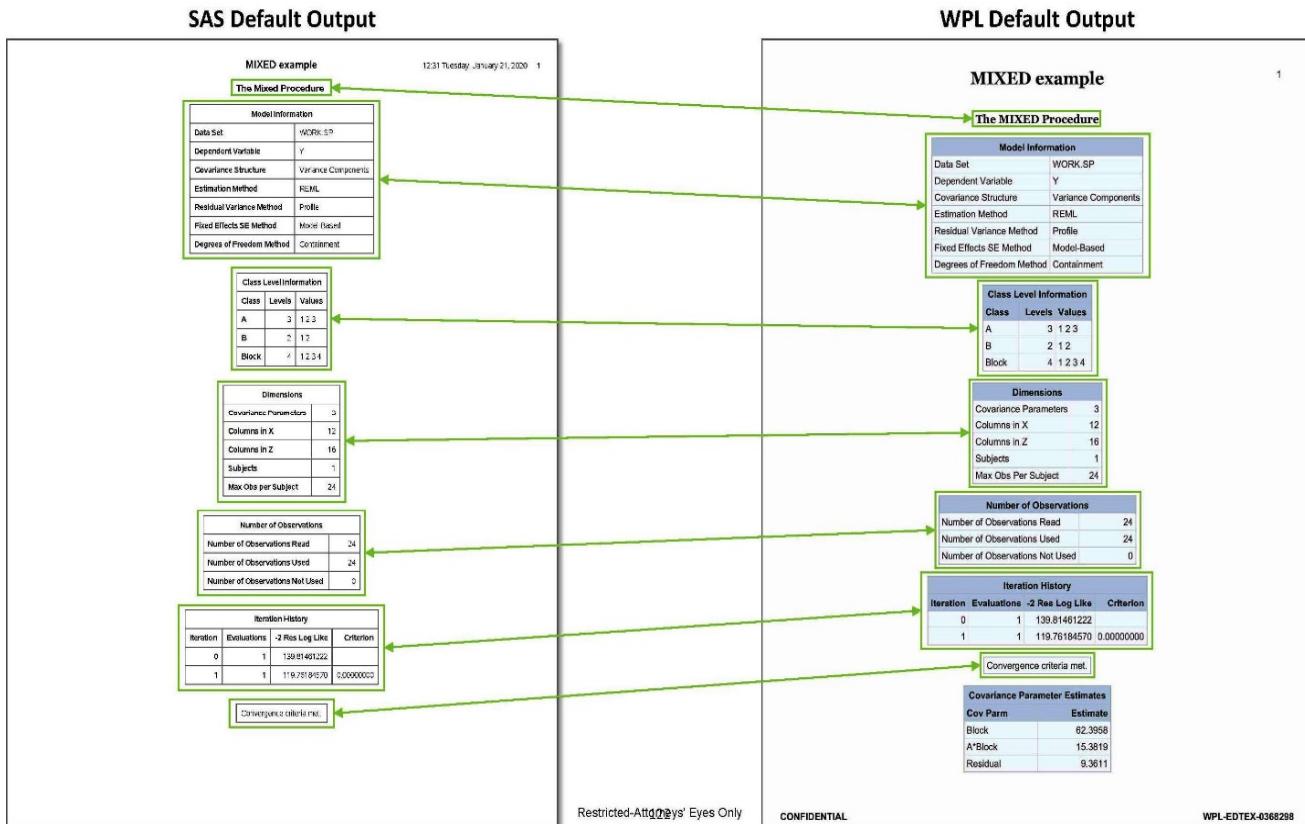
Figure 1 shows the default output design generated by SAS and WPS by the following five lines of code (which use only three of those input format elements):

REDACTED AND PUBLICLY FILED

```

56      + proc mixed;
57      +   class A B Block;
58      +   model Y = A B A*B;
59      +   random Block A*Block;
60      + run;

```

Figure 1

(Storer Decl. at ¶ 29.)

C. Creativity and Expressiveness in the SAS System.

As the court held in *Oracle*, the abstraction-filtration-comparison analysis begins with a “nuanced assessment of the particular program.” 750 F.3d 1357. The SAS System is a software product for data entry, retrieval, and management, report writing and graphics, statistical and mathematical analysis, forecasting and decision support, project management, and quality improvement. (Collins Decl. ¶ 4.) As Keith Collins, SAS’s former Chief Technical Officer, testified at deposition, the “SAS system is a collection of capabilities, primarily defined in the market by data management, analytics and business intelligence, and it is the combination of those

REDACTED AND PUBLICLY FILED

three together that enable our customers to be successful managing their data of their analytics and their information.” (Collins Tr. 11-12.)⁶ It was Collins’s job to “coordinate the development of the SAS System across the different divisions” of SAS’s research and development. (*Id.* at 42.)

The SAS System “reflects millions of man-hours of creative and difficult development and programming work on the part of thousands of SAS statisticians and programmers over the course of many years.” (Collins Decl. ¶ 4.) The company employs nearly 14,000 persons, thousands of them in R & D. (*Id.*) Many of those employees hold advanced degrees in technical subjects, including several hundred Ph.D. statisticians. (*Id.*) As a practice, SAS, regularly reinvests around 25% of its revenue into R & D annually. (*See id.*)

The most common SAS System user interface – input formats – takes the form of written instructions to the software. (Collins Decl. at ¶ 5.) Those instructions are text files containing the required instructions to the SAS System and are generally referred to as “SAS Programs.” (*Id.*)

The “heart of the SAS System” are the SAS’s PROCs which “allow the user to perform various analyses on data.” (Collins Decl. ¶ 9.) At the most basic level, as Collins explained, “[c]onceptually, a procedure is the encapsulation of any specific functionality that takes in rows and columns of data and produces output.” (Collins Tr. 14.) Analogous to the Oracle API packages (which consisted of “a number of ready-to-use JAVA programs to perform common computer functions and organized...in groups,” 750 F.3d at 1348), each of SAS’s many PROCs “is separately written and has its own design, including its own syntax, options, statements, defaults, etc.” (Collins Decl. ¶ 9.)

⁶ Exhibits herein (other than those attached to the Storer Declaration or previously filed) are attached to the Declaration of Pressly M. Millen filed herewith.

REDACTED AND PUBLICLY FILED

Every SAS “PROC corresponds to a separately called piece of prewritten, specialized software.” (Collins Decl. at ¶ 9.) In other words, as with Oracle’s “pre-written” packages, a SAS PROC, for example PROC UNIVARIATE, allows a user to perform a statistical analysis rather than write the extensive code that would otherwise be required to undertake a complex univariate analysis. (See Collins Decl. at ¶ 6 (generation of output without a PROC would “otherwise require literally hundreds of thousands of lines of code in a low-level programming language.”).)

As Collins put it, “[e]very procedure has a grammar” for which SAS “define[s] the basic tenets of a procedure; input statements, output statements, options,” and the overall “structure of [the] procedure.” (Collins Tr. 16.) It is that “grammar” that gives each PROC its “organization and structure,” (*id.*), again much like Oracle’s “elaborately organized” sets of “all the names of methods, classes interfaces, and packages,” 750 F.3d at 1351, that represented the “structure, sequence, and organization” (SSO) of “the 37 packages” at issue in *Oracle. Id.*

Collins explained that that grammar “gives it the ability for a procedure writer to understand how to take their ideas, innovation and concepts and frame it to work within the execution of the SAS System.” (Collins Tr. 17.) Thus, within each PROC are option statements “which control different functions” and the specific “capability of the procedure.” (Collins Tr. 18.) In Collins’s words, the “procedure grammar is what the PROC writer does” by compiling “a set of statements in order and execution.” (*Id* at 28.) The SAS user, through his SAS program inputs, writes what is interpreted as syntax. (*Id.*) That “specific syntax and flow is designed specific[ally] for the execution of the SAS environment.” (*Id.* at 30.) Thus, “a procedure writer writes a procedure and chooses the set of capabilities that’s designed for the execution in a specific analysis” provided by the SAS System, (*id.* at 34), akin to the “step-by-step instructions for carrying out the declared function” of an API package in *Oracle. Id.* at 1349.

REDACTED AND PUBLICLY FILED

“SAS alone determines, among other things, what statements are created, how they function, what options they have (and which are excluded), and how they are made consistent with one another.” (Collins Decl. ¶ 6.) The SAS System allows a user “to answer a question or series of questions” by processing and analyzing data and then generating “the designed output with concise written instructions that would otherwise require literally hundreds of thousands of lines of code in a low-level programming language.” (*Id.*)

Collins described how, “[w]ith respect to the input formats, developers work to define a simple set of concise commands to request a comprehensive analysis” (Collins Decl. ¶ 8), and also endorsed a Declaration from a former colleague concerning the manner in which SAS designs its input formats. (*Id.*, Ex. A.) It describes the “wide range of creative design choices” in choosing “statements and option names, syntax, and default parameters” for those inputs (*id.* at ¶ 17), including assessments of the intended user audience in terms of sectors (academic, industry, or government), as well as training and education, use of abbreviated or full-length names, use of underscores, dashes or other punctuation, expressiveness vis-à-vis the underlying idea, intuitiveness and ease of remembering, choice of default values, and finally, the manner in which the “statements, options, syntax, command structure, default values, and other elements interrelate to each other to provide a user-friendly and expressive input format.” (*Id.*)

The “overall collection” of the SAS PROCs “is a significant part of what comprises the SAS System.” (Collins Decl. ¶ 9.) That collection (of PROCs themselves comprised of statements, options, and other elements), as in *Oracle*, is “like a library,” comprised of different bookshelves with different books containing individual chapters. 750 F.3d at 1349.

With regard to output designs (which was not at issue in *Oracle*), *i.e.*, the tables, graphs, and other forms of output that the user sees, Collins described how “SAS developers make sure

REDACTED AND PUBLICLY FILED

that output is clear, consistent, and intuitive for those in the industries that use it.” (Collins Decl., ¶ 11.) He also described the implementation of the SAS Output Delivery System (ODS) in the late 1990s which “represented a complete review and rethinking of the output designs of all SAS PROCs” and “enabled every PROC developer to innovate and define the output for the PROC while abstracting output objects as tables, charts, and text to different target output like data tables, pdf, html, and printers.” (*Id.*) As a part of that process, Collins discussed the “SAS internal review board” – sometimes referred to as the “Junta” – which “spent thousands of hours working through the abstractions to enable this approach to work by considering output design details such as placement of columns, number of decimal places, number alignment, color and font, and many other details.” (*Id.*) The “ODS required years of development and greatly increased the productivity of SAS PROC developers and greatly increased the effectiveness of SAS users.” (*Id.*)

The SAS PROCs are far more than just the implementation of statistical algorithms. As Collins put it, “[y]ou could take any algorithm and implement it in the 4GL [fourth-generation] language...but that’s just the algorithm.” (Collins Tr. 35.) Instead, a “tremendous amount of what is done in the SAS System is to bring together the capability of the algorithm, the ability to scale that appropriately, to get repeatable and correct results within a standard flow and execution across procedures.” (*Id.*) In total, the SAS System represents substantial creativity and expressiveness.

D. WPL’s Open and Notorious Copying of the SAS System.

Application of the abstraction-filtration-comparison analysis also requires an understanding of what the defendant has copied. *See Oracle*, 750 F.3d at 1358-59. In this case, that copying is focused precisely on the creative expression sought by users of the SAS System – its input formats and output designs. WPL’s business proposition is that it copies the SAS System and its own technical document described how WPS was “Built to Emulate SAS”:

REDACTED AND PUBLICLY FILED

How was WPS Built to Emulate SAS

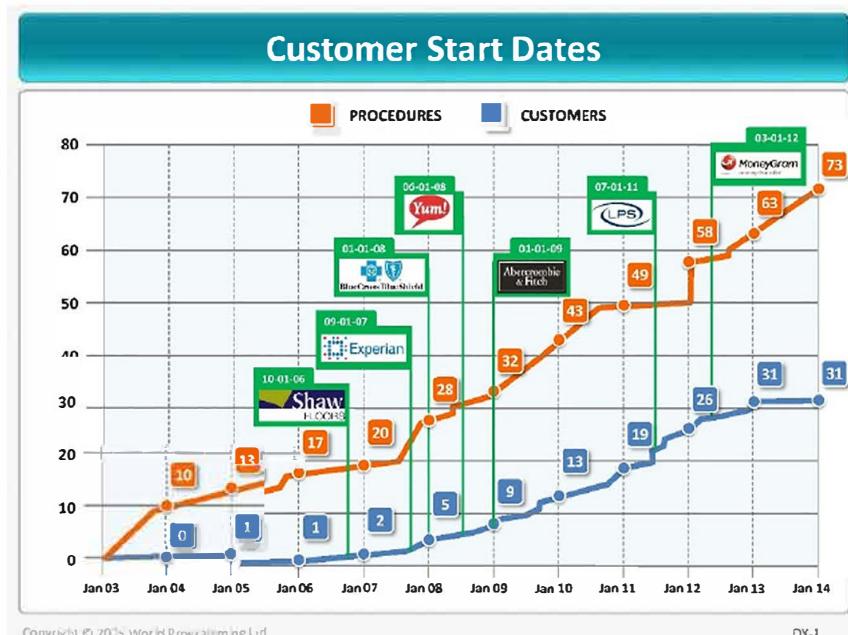
WPS has the following features in common with SAS:

- * Parses SAS Language input files
- *obeys Macro Processor rules
- * Produces similar log output
- * Produces equivalent data output
- * Produces near identical printed/readable output (Listing and HTML)
- * Reads and writes to files (DATA Step input/output)
- * Produces similar graphical output
- * Near identical CGI request processing
- * Similar RDBMS interaction
- * Equivalent but quite different Windows GUI operation

(ECF No. 264-11, World Programming System, IBM Technical Evaluation, at WPL0063723-24.)

Thus, WPL itself claims that WPS “[p]arses SAS Language input files,” *i.e.* as a whole, as well as “[p]roduces equivalent data output.” (*Id.*) Necessarily, WPL has been prolific and constant in its copying of the SAS System. Figure 2, for example, taken from a WPL trial demonstrative, shows that WPS went from copying 10 SAS PROCs in 2004 to 73 copied PROCs ten years later (a process that WPL has continued in later years).

Figure 2



REDACTED AND PUBLICLY FILED



Indeed, the extent of WPL's copying of the SAS System input formats was a commercial selling point publicly touted by WPL. For example, as of October 3, 2016, WPL published online a "Quick Reference for Language Support Concerning Statistical Analysis," for WPS Version 3.2.3 which "lists the core statistical procedure from the language of SAS supported in the WPS Core module and the additional statistical procedure from the language of SAS supported in the WPS Statistics module. (Millen Decl., Ex. C.)⁷ That document, over 50 pages long, lists, for multiple SAS PROCs, all of the Statements, and Options that were "supported" in WPS as of that date as denoted by a "Y" (meaning supported) or a blank (meaning not supported). The 50-plus pages of the "Quick Reference" show the extensive copying by WPL. (*See id.*) Thus, for example, a single page from that document (shown in Figure 3) sets out for SAS's PROC ANOVA nine Statements and 42 Options of which a total of 35 each of which are marked as "Supported." (Each individual page shows dozens and dozens of SAS elements copied into WPS.)

⁷ [REDACTED]

A large black redaction bar covers the bottom portion of the page content, obscuring the footer area.

REDACTED AND PUBLICLY FILED

Figure 3

Procedure	Statements	Options	Supported
ACECLUS			Y
ANOVA	Proc ANOVA		Y
		DATA=	Y
		MANOVA	Y
		MULTIPASS	Y
		NAMELEN=	Y
		NOPRINT	Y
		ORDER=	Y
		OUTSTAT=	Y
		PLOTS	Y
	ABSORB		Y
	BY	DECENDING	Y
		NOTSORTED	Y
	CLASS	TRUNCATE	Y
	FREQ		Y
	FORMAT		Y
	INFORMAT		Y
	LABEL		Y
	MANOVA	CANONICAL	
		E=	
		H=	
		M=	
		MINAMES=	
		MSTAT=	
		ORTH	
		PREFIX=	
		PRINTH	
		SUMMARY	
	MEANS	ALPHA=	Y
		BON	Y
		CLDIFF	Y
		CLM	Y
		DUNCAN	Y
		DUNNETT	Y
		DUNNETTL	Y
		DUNNETTU	Y
		E=	
		GABRIEL	Y
		GT2	Y
		HOVETEST=	
		KRATIO	
		LINES	Y
		LSD	Y
		NOSORT	Y
		REGWQ	Y
		SCHEFFE	Y
		SIDAK	Y
		SMM	Y
		SNK	Y

(*Id.* at 00474244.)

Over time, WPL has continued to implement an ever-expanding list of the SAS System input formats and output designs in WPS.

Copying of SAS's output designs was similarly crucial to the commercial proposition of WPS. As one WPL employee testified, “[a]ll users of the SAS software expect WPS to give *precisely the same output* as is produced by the SAS software in response to any given input. Any deviation in the output from WPS as compared to that produced by the SAS software is perceived

REDACTED AND PUBLICLY FILED

by end users to be a WPS bug.” (ECF No. 264-18, at ¶ 15, emphasis added.) A WPL contractor, also in sworn testimony, added “[n]ot only do the numbers have to be mathematically correct, *any report layout also has to be the same.*” (ECF No. 264-19, at ¶ 52 (emphasis added).)

With regard to output designs, SAS has come forward with numerous examples of near-identical output designs in WPS with respect to given SAS PROCs that have been copied in WPS. (Storer Decl., Exs. D, E & F.) SAS has also provided examples showing that the input formats and output designs for analogous statistical analyses in other companies’ software are different.

V. WPL’S ABSTRACTION-FILTRATION PROCESS IS UNAVAILING.

A. The Burden of Further Filtration is on WPL.

Rather than attempting to conduct the Abstraction-Filtration-Comparison analysis itself, WPL and its expert instead lodge various criticisms of SAS’s proposed A-F-C analysis, advancing, all at once, arguments that would filter out everything from the SAS System based on (1) public domain, (2) merger, (3) *scènes à faire*, and (4) short phrases. At the same time, WPL does not ever propose an alternate way for the Court to tease out the elements of the SAS System or to filter those various components. Recent guidance from the Eleventh Circuit on the issue of burden, *Compulife Software Inc. v. Newman*, 959 F.3d 1288 (11th Cir. 2020), makes clear that WPL has fundamentally failed to carry its burden by doing so. In *Compulife*, the appeals court reversed a district court which, among other errors, “improperly placed the burden on [plaintiff] to prove, as part of the filtration analysis, that the elements the defendants copied were protectable.” *Id.* at 1303. Instead, the court held, the district court “should have required the defendants to prove that those elements were *not* protectable.” *Id.* To do anything else ultimately puts the plaintiff to the near-impossible task of “prov[ing] a negative” in the context of a filtration analysis. *Id.* at 1305.

The court noted that “[f]iltration can be tricky because copied material may be unprotectable for a wide variety of reasons,” 959 F.3d at 1304, including, as discussed in *Oracle*,

REDACTED AND PUBLICLY FILED

exclusion under 17 U.S.C. § 102, merger, public domain, and *scènes à faire*. *Id.* The court clarified its position that “after an infringement plaintiff has demonstrated that he holds a valid copyright and that the defendant engaged in factual copying, the *defendant* bears the burden of proving – as part of the filtration analysis – that the elements he copied from a copyrighted work are *unprotectable*.” *Id.* at 1305 (emphasis in original). The court ruled that such an approach was “consonant with its existing precedent and it enjoys the support of the foremost copyright treatise.” *Id.*⁸ In further explaining why a plaintiff should not be required “to prove a negative,” the court explained that “[i]f the plaintiff had the burden of proving protectability, he would have to preemptively present evidence negating all possible theories of unprotectability just to survive a motion for summary judgment.” *Id.* The court summarized:

A plaintiff, for instance, can’t be expected to present the entirety of the public domain as it existed when he authored his copyrighted material in order to show that no elements of his work were taken from it. Nor could a plaintiff reasonably introduce the entire corpus of relevant, industry-standard techniques just to prove that none of the material copied from his work constituted *scènes à faire*. Placing the burden of proving protectability on the plaintiff would seemingly require just these kinds of impossibilities.

Id. at 1305-06.

The burden here, therefore, is on WPL. If it can “demonstrate[] – at the filtration stage – that it copied only unprotectable material, such that no substantial similarities remain after filtration, the defendant is entitled to summary judgment.” 959 F.3d at 1306. Here, though, WPL has never claimed that it only copied merger, public domain, or *scènes à faire* material. Instead, WPL copied extensively from the SAS System input formats and output designs, and, in this case, has mounted a consistent critique that SAS is required to “prove the negative” disclaimed by the court in *Compulife*.

⁸ Citing 4 Nimmer on Copyright § 13.03[F][3].

REDACTED AND PUBLICLY FILED

B. WPL’s Filtration Analysis Is Wrong.

WPL’s copyrightability arguments take a blunderbuss approach, advancing, all at once, arguments that would filter out everything from the SAS System based on (1) public domain, (2) merger, (3) *scènes à faire*, and (4) short phrases. Despite having copied a massive amount of the SAS System, WPL makes no attempt to filter out what it claims are the unprotected parts of the SAS System from the creative expression that is protectable. In the end, *Oracle* and the other pertinent authorities show why WPL’s approach is fundamentally flawed.

1. Public Domain

WPL’s public domain argument is a rehash of an argument raised by another infringer which failed over 35 years ago. *SAS Inst. Inc. v. S & H Computer Systems, Inc.*, 605 F.Supp. 816 (M.D. Tenn. 1985). In that case, SAS prevailed on its claims concerning the copying of a version of SAS referred to as SAS 79.5 (released in 1982). *Id.* at 833. That ruling came despite the same court’s previous ruling that an earlier version of the SAS software was in the public domain. *Id.* at 827. The court, in rejecting the public domain argument, held that “SAS 79.5 represents *overwhelmingly* a new and original work of authorship, above and beyond the pre-existing work contained in earlier release [sic] of SAS” and that therefore, [a]ccordingly, the copyright in SAS 79.5 is valid and fully enforceable.” *Id.* (emphasis added).

Even WPL’s own expert conceded that PROCs from the earlier public domain version had “all changed” since the 1970s (ECF No. 296-2, at 190), with “additional language elements added that would let you do – or perform different, in some cases, analysis.” (*Id.* at 191.) When asked whether the outputs available in 1976 were more primitive than those available in 2020, WPL’s expert stated that the “range of outputs and the nature of the outputs certainly was not – didn’t allow a user the kind of sophisticated control or the ability to generate what we consider today to

REDACTED AND PUBLICLY FILED

be high quality graphics.” (*Id.* at 194.) In other words, the output designs have markedly changed since any public domain determination regarding a 44 year-old version of the SAS System.

Professor Storer’s Expert Report explained why he did not filter out public domain material (*see* Storer Rep. at 33), and his rationales mirrored those of the court in *S & H Computer*. *See* 605 F.Supp. at 819 (“The Court finds that SAS 79.5 contained numerous and substantial additions, enhancements, revisions, and other new material not contained in SAS 76.2.”). The same is even more true decades after the ruling in *S & H Computer*.

Significantly, WPL does not claim to have started with the rudimentary input formats and primitive outputs designs available in 1976 and then developed its own original work of authorship by adding to, enhancing, and revising that early version of SAS with its own creative expression. Instead, the record reflects an exhaustive effort by WPL to copy the SAS System as offered for sale to SAS’s customers today. If WPL’s position is that certain PROC names or other elements should be filtered out of the SAS System – and are not protected in any event as part of the combination of input formats in the SAS System, as in *Cisco* – WPL has not even attempted a filtration analysis based on the elements in the modern SAS System it claims must be filtered out (or that it has specifically copied) as part of the Abstraction-Filtration-Comparison analysis.

2. Merger

With regard to the merger doctrine, the court in *Oracle* stated that copyright law would “not protect a copyrighted work from infringement if the idea contained therein can be expressed in only one way.” 750 F.3d at 1360. However, the “unique arrangement of computer program expression ... does not merge with the process so long as alternate expressions are available.” *Id.* (quoting *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 840 (Fed. Cir. 1992)). In ruling that the merger doctrine did not apply, the court stated that the “evidence showed that Oracle had unlimited options as to the selection and arrangement.” 750 F.3d at 1361 (internal quotation

REDACTED AND PUBLICLY FILED

omitted). Using the “java.lang.Math.max” example (*see* n. 4, *supra*), the court accepted Oracle’s explanation that “the developers could have called it any number of things, including ‘Math.Maximum’ or ‘Arith.larger.’” *Id.* Thus, the court held, “[t]his was not a situation where Oracle was selecting among preordained names and phrases to create its packages.” *Id.*

Here, as in *Oracle*, “alternate expressions [were] available” to SAS. *Oracle*, 750 F.3d at 1360. Professor Storer provided examples of alternate expressions from competing software packages (*see* Storer Rep. at 34-36), and WPL has come forward with no evidence that SAS developers were in any way limited in their expressive choices by a lack of alternative expressions. As in *Oracle*, SAS developers “had unlimited options as to the selection and arrangement” of the input formats and output designs. 750 F.3d at 1361.

3. *Scènes à faire*

As with merger, WPL has come forward with no evidence that the expression sought to be protected by SAS embodies some sort of industry standard. While the doctrine might deny protection to “program elements” that are “dictated by external factors” like “widely accepted programming practices within the computer industry” or dictated by the “mechanical specifications of the computer,” *Oracle*, 750 F.3d at 1353, 1363, there is no evidence here that anything asserted by SAS falls into those categories.

4. Short Phrases

In *Oracle*, the Federal Circuit also rejected the district court’s handling of the issue of “short phrases,” finding that “by dissecting the individual lines of declaring code at issue into short phrases, the district court further failed to recognize that an original combination of elements can be copyrightable.” 750 F.3d at 1362. In other words, the district court “failed to recognize...that the relevant question for copyrightability purposes is not whether the work at issue contains short

REDACTED AND PUBLICLY FILED

phrases – as literary works often do – but, rather, whether those phrases are creative.” *Id.* The court provided an “analogy,” equally apt here:

the opening of Charles Dickens’ *A Tale of Two Cities* is nothing but a string of short phrases. Yet no one could contend that this portion of Dickens’ work is unworthy of copyright protection because it can be broken into those shorter constituent components. The question is not whether a short phrase or series of short phrases can be extracted from the work, but whether the manner in which they are used or strung together exhibits creativity.

Id. at 1363.

The Federal Circuit, therefore, ultimately concluded that the district court erred when it “apparently focused on individual lines of code,” because “Oracle is not seeking copyright protection for a specific short phrase or word.” 750 F.3d at 1353. Instead, “[b]ecause Oracle ‘exercised creativity in the selection and arrangement’ of the method declarations when it created the API packages and wrote the relevant declaring code, they contain protectable expression that is entitled to copyright protection.” *Id.* (quoting *Atari*, 975 F.2d at 840). Accordingly, the Federal Circuit “conclude[d] that the district court erred in applying the short phrases doctrine to find the declaring code not copyrightable.” *Id.*

WPL’s assertion of the concept of “short phrases” seeks to rely upon a method discredited by the Federal Circuit in *Oracle* “by dissecting the individual lines of [the input formats] at issue into short phrases,” while “fail[ing] to recognize that an original combination of elements can be copyrightable.” 750 F.3d at 1362. In other words, as held in *Oracle*, WPL may not “fail[] to recognize...that the relevant question for copyrightability purposes is not whether the work at issue contains short phrases – as literary works often do – but, rather, whether those phrases are creative.” *Id.* The issue, in other words, is not how “short” a phrase may be, but “whether the phrase contains some appreciable level of creativity, however few words it may contain.” *J. Racenstein & Co. v. Wallace*, 1999 WL 632853, at *1-2 (S.D.N.Y. Aug. 19, 1999).

REDACTED AND PUBLICLY FILED

Equally important, as noted in *Oracle*, Oracle was “not seeking copyright protection for a specific short phrase or word,” 750 F.3d at 1353. Here too, SAS is not seeking copyright protection for each one of the thousands of expressions copied by WPL, e.g., the “anovaF” option in PROC MIXED, but instead seeks to demonstrate that it “exercised creativity in the selection and arrangement” in the SAS input formats as a whole. *Id.* (quoting *Atari*, 975 F.2d at 840). Simply “because words such as ‘minimum,’ ‘optimum,’ ‘natural,’ ‘line,’ and the like are, in of themselves, uncopyrightable, it does not necessarily follow that a particular arrangement of those words to express an idea in a particular manner is uncopyrightable.” *Compaq Computer Corp. v. Ergonomics, Inc.*, 137 F.Supp.2d 768, 774-75 (S.D. Tex. 1981). The SAS expressions, in other words, do not exist in isolation, but instead as part of the overall arrangement of the various PROCs thus taking “their meaning from the context of the whole and serv[ing] the same purpose as the copyrighted work.” *Ventures Educ. Sys. Corp. v. Prof'l Dev. Assocs., Inc.*, 2008 WL 316667, at *3 (S.D.N.Y. Jul. 31, 2008); see also *Cisco Sys., Inc.*, slip op. at 17-18.

VI. JUDICIAL RESOURCES MAY BE WASTED IF THE COURT OR JURY RESOLVES ISSUES CONCERNING SAS’S INPUT FORMATS BEFORE THE SUPREME COURT ANNOUNCES ITS OPINION IN *GOOGLE LLC V. ORACLE AMERICA INC.*

SAS alleges in this lawsuit that WPL copied the SAS System including its collections of input formats and output designs. Following discovery, the parties cross-moved for summary judgment on the copyrightability of those formats and designs. The Court then denied both motions without prejudice, explaining that “a ruling on [the] competing motions for summary judgment within the constraints of Rule 56” could “set us all up for a second trial at a later date, which is a waste of time and resources.” (Aug. 24, 2020, Pretrial Hr’g Tr. at 23:1–24:3.) The Court continued the trial until January 2021 and ordered additional briefing and testimony on copyrightability because a more accelerated procedure creates a higher “risk of the jury getting

REDACTED AND PUBLICLY FILED

into a situation where either [the Court has] to order a new trial or this case becomes reversible on appeal, and none of us want that.” (*Id.* at 18:9–11.)

SAS had originally anticipated the earlier trial date and remains eager to present its case to a jury in January 2021. But, given the Court’s comments and the change in the trial date, SAS is compelled to point out that the Supreme Court will hear oral argument on October 7, 2020 (a date now prior to the current trial date), in *Google LLC v. Oracle America Inc.*, No. 18-956 (an appeal of a case extensively cited herein). The Supreme Court is expected to issue an opinion in that case at some point between January 2021 (three months after oral argument) and June 2021 (the end of the term).⁹ Regardless of the outcome, the Supreme Court’s decision will provide guidance on the proper scope of (and test for evaluating) copyright protection for software input formats – one of the central issues in this case. The first question presented is “[w]hether copyright protection extends to a software interface,” which the petitioner defined as “lines of computer code that allow developers to operate prewritten libraries of code used to perform particular tasks.” (Pet. Writ Cert. at I, *Google LLC v. Oracle America, Inc.*, 2019 WL 338902 (U.S. Jan. 24, 2019).) The input formats and SSO at issue in the Supreme Court are analogous to the SAS input formats and SSO in dispute here. (See § IV, *supra*.) As the United States explains as amicus curiae supporting the respondent, Google “copied respondent’s declaring code verbatim, while writing its own implementing code.” (Br. United States at 7, 2020 WL 1028353, at *7 (U.S. Feb. 19, 2020).)

As explained above (at § IV, *supra*), the same issue is present here. The Supreme Court’s acceptance or rejection of Google’s argument regarding the copyrightability of that declaring code

⁹ Cf. Lee Epstein et al., *The Best for Last: The Timing of U.S. Supreme Court Decisions*, 64 Duke L.J. 991, 993 (2015) (“The Court usually issues its decision within three months of oral argument; only the very rare case bounces around the chambers for more than six months, and almost every decision is issued in the same term in which the case is argued.” (footnotes omitted)).

REDACTED AND PUBLICLY FILED

and SSO is likely to control much of input side of this case.¹⁰ Given the Court’s concerns for efficient use of judicial resources, SAS would not oppose a short stay of this case pending the release of that decision later this term.

Relatedly, SAS files this brief in anticipation of a copyrightability hearing without waiving its Seventh Amendment demand for “a jury trial on all issues so triable.” (SAS’s Am. Compl. at 67, ECF No. 128.) In scheduling that copyrightability hearing, the Court noted that “[v]arious authorities hold that copyrightability is at least in part a question of law, reserved for determination by the Court.” (Order As To Copyrightability at 1.) The authorities cited by the Court are collected in footnote 3 of the Federal Circuit’s decision in *Oracle*, 750 F.3d at 1353. In that footnote, the Federal Circuit acknowledged that the “Supreme Court has not addressed whether copyrightability is a pure question of law or a mixed question of law and fact, or whether, if it is a mixed question of law and fact, the factual components of that inquiry are for the court, rather than the jury.” *Id.* at 1353 n.3. The Federal Circuit did not resolve that question “because the parties . . . agreed that the district court would decide copyrightability.” *Id.*

Unlike the parties in *Oracle*, SAS preserves its argument that copyrightability is a mixed question of law and fact that should be submitted to the jury as part of its ultimate decision on infringement (subject, of course, to the typical standards governing summary judgment). Under current law, copyright protection for computer programs requires the factfinder to filter out the “unprotectable elements suggested by the *particular facts* of the program under examination.” *Eng’g Dynamics*, 26 F.3d at 1343 (emphasis added) (quoting *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 834 (10th Cir. 1993)). In *Engineering Dynamics*, the particular facts

¹⁰ A second aspect of this case, concerning WPL’s copying of SAS’s output designs, is not before the Supreme Court.

REDACTED AND PUBLICLY FILED

included the “competing structural design programs [that] exist in the market,” *id.* at 1345, and the “question [was] whether the utilitarian function of the input formats . . . outweigh their expressive purpose so as to preclude copyright protection,” *id.* at 1346.

This “application-of-legal-standard-to-fact sort of question . . . , commonly called a ‘mixed question of law and fact,’ has typically been resolved by juries.” *United States v. Gaudin*, 515 U.S. 506, 512 (1995). The Fifth Circuit recognized as much in *Engineering Dynamics* when it remanded the case to the factfinder (there, the district court) because of “the factual content of many of these issues.” 26 F.3d at 1343. Similarly, the Fifth Circuit held in *Aspen Tech., Inc. v. M3 Tech., Inc.*, 569 F. App’x 259, 270 (5th Cir. 2014) (per curiam), that the “jury, as the ultimate factfinder, was entitled to determine whether the copied aspects of the program were entitled to copyright protection.” *Accord N. Coast Indus. v. Jason Maxwell, Inc.*, 972 F.2d 1031, 1035 (9th Cir. 1992) (“The plaintiff was entitled to have the validity of its copyright determined by a trier-of-fact.”).¹¹

Dated: September 7, 2020

Respectfully submitted,

/s/ Pressly M. Millen
 Pressly M. Millen
Pro Hac Vice
 Raymond M. Bennett
Pro Hac Vice
 Samuel B. Hartzell
Pro Hac Vice
 WOMBLE BOND DICKINSON (US) LLP
 555 Fayetteville Street, Suite 1100
 Raleigh, North Carolina 27601
 (919) 755-2135
 press.millen@wbd-us.com
 ray.bennett@wbd-us.com
 sam.hartzell@wbd-us.com

¹¹ The factual underpinnings of copyrightability are unlike the issues presented at claim construction hearings, where the trial court is engaged in the “construction of written instruments.” *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 388 (1996).

REDACTED AND PUBLICLY FILED

Christian E. Mammen
Pro Hac Vice
Carrie Richey
WOMBLE BOND DICKINSON (US) LLP
1841 Page Mill Road, Suite 200
Palo Alto, California 94304
(408) 341-3067
chris.mammen@wbd-us.com
carrie.richey@wbd-us.com

Michael C. Smith
State Bar No. 18650410
SIEBMAN, FORREST, BURG & SMITH,
LLP
113 East Austin Street
Marshall, Texas 75671
(903) 938-8900
michaelsmith@siebman.com

Attorneys for Plaintiff SAS Institute Inc.

REDACTED AND PUBLICLY FILED

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document was served on counsel for Defendants via the Court's CM/ECF system and by email on September 7, 2020.

/s/ Pressly M. Millen
Pressly M. Millen